CLAIMS:

1. A method of executing processor tasks on a multi-processing system, the multi-processing system including a plurality of processing units that may access a shared memory, the method comprising:

providing that selected processor tasks be copied from the shared memory and executed by one or more of the processing units; and

migrating at least one processor task from one of the processing units to another of the processing units.

2. The method of claim 1, further comprising prohibiting the execution of the processor task from the shared memory.

3. The method of claim 1, wherein the plurality of processing units comprises a main processor unit and a plurality of sub-processing units, each of the plurality of sub-processing units having a local memory, and wherein the processor tasks are copied to local memory and executed in local memory.

4. The method of claim 1, wherein migration of the at least one processor task is based on a condition.

5. The method of claim 4, wherein the condition is based on respective priority levels associated with the processor tasks.

6. The method of claim 5, wherein satisfaction of the condition and the initiating of the migration is not based on preemptive action.

7. The method of claim 5, further comprising: requiring that the sub-processing units select processor tasks from the shared memory for execution based on their priority levels.

8. The method of claim 3, further comprising: requiring that the sub-processing units select a processor

task of higher priority before a processor task of lower priority from the shared memory.

9. The method of claim 3, further comprising:

selecting a first processor task of a first priority level from the shared memory for execution by a first sub-processing unit;

selecting a second processor task of a second priority level from the shared memory for execution by a second sub-processing unit; and

yielding the first sub-processing unit to a third processor task of a third priority level, the third processor task being selected because its priority level is higher than any other processor tasks that a ready to be executed.

10. A method of executing processor tasks on a multi-processing system, the multi-processing system including a plurality of processing units that may access a shared memory, the method comprising:

providing that selected processor tasks be copied from the shared memory and executed by one or more of the processing units;

providing that the processing units select processor tasks from the shared memory for execution based on priority levels of the processor tasks; and

providing that a processor task of lower priority running on one of the processing units may be preemptively replaced with a processor task of higher priority.

11. The method of claim 10, further comprising prohibiting the execution of the processor task from the shared memory.

12. The method of claim 10, wherein the plurality of processing units comprises a main processor unit and a plurality of sub-processing units, each of the plurality of sub-processing units having a local memory, and wherein the

processor tasks are copied to local memory and executed in local memory.

13. The method of claim 12, further comprising: requiring that the sub-processing units select a processor task of higher priority before a processor task of lower priority from the shared memory.

14. The method of claim 12, further comprising:

selecting a plurality of processor tasks of associated priority levels from the shared memory for execution by a number of sub-processing units;

causing an n-th processor task in the shared memory having a given priority level to become ready for execution; and

determining whether the given priority level is higher than any of the priority levels of the plurality of processor tasks.

15. The method of claim 14, wherein at least one of the sub-processing units is operable to perform the determination.

16. The method of claim 14, further comprising: preemptively replacing one of the plurality of processor tasks of lower priority level than the given priority level with the n-th processor task.

17. The method of claim 16, wherein one or more of the sub-processing units is operable to at least initiate the replacement and cause the one of the plurality of sub-processing units to yield execution of the processor task of lower priority level.

18. The method of claim 17, further comprising: providing that the initiating sub-processing unit issues an interrupt to the yielding sub-processing unit in order to initiate the replacement of the processor task of lower priority level.

19. The method of claim 17, further comprising: providing that the yielding sub-processing unit writes the

processor task of lower priority from its local memory back into the shared memory.

20. The method of claim 17, further comprising: providing that the yielding sub-processing unit copies the n-th processor task of higher priority from the shared memory into its local memory for execution.

21. A method of executing processor tasks on a multi-processing system, the multi-processing system including a plurality of sub-processing units and a main processing unit that may access a shared memory, each sub-processing unit including an on-chip local memory separate from the shared memory, the method comprising:

providing that the processor tasks be copied from the shared memory into the local memory of the sub-processing units in order to execute them, and prohibiting the execution of the processor tasks from the shared memory;

selecting a plurality of processor tasks of associated priority levels from the shared memory for execution by a number of sub-processing units;

providing that the sub-processing units may determine whether an n-th processor task in the shared memory having a given priority level is higher than any of the priority levels of the plurality of processor tasks.

22. The method of claim 21, further comprising: providing that a processor task of lower priority running on one of the sub-processing units may be preemptively replaced with a processor task of higher priority.

23. The method of claim 21, further comprising: providing that the sub-processing units use a shared task priority table in determining whether the n-th processor task is of a higher priority level than the plurality of processor tasks.

24. The method of claim 23, wherein:

the shared task priority table includes entries of for sub-processing unit identifiers and processor task priority identifiers; and

each entry includes a sub-processing unit identifier and priority identifier pair that indicate a priority level of a given processor task running on an associated sub-processing unit.

25. The method of claim 21, further comprising: providing that a sub-processing unit seeking to determine whether the n-th processor task is of a higher priority level than the plurality of processor tasks searches the shared task priority table to find an entry pair indicating a lower priority level.

26. A method of executing processor tasks on a multi-processing system, the multi-processing system including a plurality of sub-processing units and a main processing unit that may access a shared memory, each processing unit including an on-chip local memory separate from the shared memory, the method comprising:

providing that the processor tasks be copied from the shared memory into the local memory of the sub-processing units in order to execute them, and prohibiting the execution of the processor tasks from the shared memory;

providing that the sub-processing units select processor tasks from the shared memory for execution based on priority levels of the processor tasks; and

migrating a processor task of higher priority running on a given one of the sub-processing units to another of the sub-processing units running a processor task of lower priority in response to an interrupt received by the given sub-processing unit.

27. A multi-processor apparatus, comprising:

a plurality of processing units, each processing unit including local memory in which to execute processor tasks; and

a shared memory operable to store processor tasks that are ready to be executed, wherein:

the processor tasks are copied from the shared memory into the local memory of the processing units to execute them, and

at least one processor task is migrated from one of the processing units to another of the processing units.

28. The apparatus of claim 27, further comprising prohibiting the execution of the processor task from the shared memory.

29. The method of claim 27, wherein the plurality of processing units comprises a main processor unit and a plurality of sub-processing units, each of the plurality of sub-processing units having a local memory, and wherein the processor tasks are copies to local memory and executed in local memory.

30. The apparatus of claim 29, wherein migration of the at least one processor task is based on a condition.

31. The apparatus of claim 30, wherein the condition is based on respective priority levels associated with the processor tasks.

32. The apparatus of claim 31, wherein satisfaction of the condition and the initiating of the migration is not based on preemptive action.

33. The apparatus of claim 31, wherein the sub-processing units are operable to select processor tasks from the shared memory for execution based on their priority levels.

34. The apparatus of claim 29, wherein the sub-processing units are operable to select a processor task

of higher priority before a processor task of lower priority from the shared memory.

35. The apparatus of claim 29, wherein:

a first sub-processing unit is operable to select a first processor task of a first priority level from the shared memory for execution;

a second sub-processing unit is operable to select a second processor task of a second priority level from the shared memory for execution; and

the first sub-processing unit is operable to yield to a third processor task of a third priority level, the third processor task being selected because its priority level is higher than any other processor tasks that a ready to be executed.

36. The apparatus of claim 29, wherein the sub-processing units are operable to:

select a plurality of processor tasks of associated priority levels from the shared memory for execution; and

determine whether an n-th processor task in the shared memory having a given priority level that has become ready for execution has a higher level priority than any of the priority levels of the plurality of processor tasks.

37. The apparatus of claim 36, wherein at least one of the sub-processing units is operable to perform the determination.

38. The apparatus of claim 36, wherein at least one of the sub-processing units is operable to preemptively replace one of the plurality of processor tasks of lower priority level than the given priority level with the n-th processor task.

39. The apparatus of claim 38, wherein one or more of the sub-processing units is operable to at least initiate the replacement and cause the one of the plurality of

sub-processing units to yield execution of the processor task of lower priority level.

40. The apparatus of claim 39, wherein the initiating sub-processing unit is operable to issue an interrupt to the yielding sub-processing unit in order to initiate the replacement of the processor task of lower priority level.

41. The apparatus of claim 39, wherein the yielding sub-processing unit is operable to write the processor task of lower priority from its local memory back into the shared memory.

42. The apparatus of claim 39, wherein the yielding sub-processing unit is operable to copy the n-th processor task of higher priority from the shared memory into its local memory for execution.

43. A multi-processor apparatus, comprising:

a plurality of sub-processing units, each sub-processing unit including an on-chip local memory in which to execute processor tasks; and

a shared memory operable to store processor tasks that are ready to be executed, wherein:

the processor tasks are copied from the shared memory into the local memory of the sub-processing units in order to execute them, and the processor tasks are not executed from the shared memory,

the sub-processing units are operable to select processor tasks from the shared memory for execution based on priority levels of the processor tasks; and

at least one of the sub-processing units is operable to migrate a processor task of higher priority running on a given one of the sub-processing units to another of the sub-processing units running a processor task of lower priority in response to an interrupt received by the given sub-processing unit.

44. The apparatus of claim 43, wherein the sub-processing units are operable to select a processor task of higher priority before a processor task of lower priority from the shared memory.

45. The apparatus of claim 43, wherein the sub-processing units are operable to:

select a plurality of processor tasks of associated priority levels from the shared memory for execution; and

determine which of the plurality of processor tasks running on the of sub-processing units has a lowest priority level that is lower than the priority level of the processor task running on the given sub-processing unit.

46. The apparatus of claim 45, wherein the given sub-processing unit is operable to perform the determination.

47. The apparatus of claim 45, wherein the given processor task is migrated to the sub-processing unit running the processor task of lowest priority level and replacing that processor task.

48. The apparatus of claim 47, wherein the given sub-processing unit is operable to at least initiate the migration and causing the sub-processing unit running the processor task of lowest priority level to yield execution to the given processor task of higher priority level.

49. The apparatus of claim 48, wherein the given sub-processing unit is operable to issue an interrupt to the yielding sub-processing unit in order to initiate the replacement of the processor task of lowest priority level.

50. The apparatus of claim 48, wherein the yielding sub-processing unit is operable to write the processor task of lower priority from its local memory back into the shared memory.

51. The apparatus of claim 45, wherein the yielding sub-processing unit is operable to copy the given processor task

of higher priority from the local memory of the given sub-processing unit into its local memory for execution.

52. The apparatus of claim 43, wherein the given sub-processing unit is operable to use a shared task priority table in determining which processor task is of the lowest priority level.

53. The apparatus of claim 52, wherein:

the shared task priority table includes entries of for sub-processing unit identifiers and processor task priority identifiers; and

each entry includes a sub-processing unit identifier and priority identifier pair that indicate a priority level of a given processor task running on an associated sub-processing unit.

54. The apparatus of claim 53, wherein the given sub-processing unit is operable to search the shared task priority table to find an entry pair indicating a lowest priority level.

55. The apparatus of claim 53, wherein the sub-processing units are operable to modify the shared task priority table such that the entry pairs are current.